AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application.

LISTING OF CLAIMS:

1. (Original) A method for computing distances between a received point and four points in a two-dimensional grid with a constellation representing a number of bits greater than three, wherein each of the four points belong to a unique coset in the constellation, the method comprising:

determining a first point on a grid nearest to the received point;

computing a second point closest to the received point inside a specified area;

computing a third, fourth, and fifth point, wherein each point is a member of a different coset and each point is the closest point in its coset to the received point; and

computing a distance from the received point to each of the second, third, fourth, and fifth points.

2. (Original) The method of claim 1 further comprising after the first computing, recomputing the second point if the second point is invalid.

3. (Original) The method of claim 2, wherein the second point is invalid if it is outside of the constellation.

4. (Original) The method of claim 1, wherein the first point can be determined by evaluating:

round((Rx + iRy – 1 – i)/2*2 + 1 + I,

wherein Rx and Ry are two-dimensional components of the received point, i is the

imaginary number, and round(.) is an operator that returns an integer number closest to a value provided to it.

5. (Original) The method of claim 1, wherein the number of bits is an even value, wherein the received point can be expressed in two-dimensional components Rx and Ry, and wherein the first computing comprises:

determining if Rx and Ry lie inside a square specified by the number of bits; and

computing two-dimensional components of the second point based on the second determining.

6. (Original) The method of claim 5, wherein the second determining comprises:

setting Cx = 1 if Rx lies inside a boundary of the square, else Cx = -1;

setting Cy = 1 if Ry lies inside a boundary of the square, else Cy = -1;

and wherein the fourth computing comprises

setting Ax = sign(RGx) * MAX_{XY} if Cx = -1, else Ax = RGx; and

setting Ay = sign(RGy) * MAX_{XY} if Cy = -1, else Ay = RGy,

wherein Ax and Ay are two-dimensional components of the second point, RGx and RGy are two-dimensional components of the first point, MAX_{XY} is value describing the size of the square and can be computed by $2^{\text{number of bits}/2} - 1$.

7. (Original) The method of claim 5, wherein the second computing comprises:

computing an intermediate value, d, wherein d = the received point – the second point;

setting the third point = the second point + Cx * sign(dx) * 2;

setting the fourth point = the second point + i * Cy * sign(dy) * 2; and

setting the fifth point = the second point + 2(Cx * sign(dx) + i * Cy * sign(dy)), wherein Cx and Cy are values specifying if the two-dimensional components of the received point lie inside a boundary of the square and dx and dy are two-dimensional components of d.

8. (Original) The method of claim 5, wherein the third computing comprises computing a Euclidean distance from the received point to each of the second, third, fourth, and fifth points.

9. (Original) The method of claim 8, wherein each of the second, third, fourth, and fifth points belong to a unique coset.

10 – 22 Cancelled

23. (Original) The method of claim 1, wherein the method can be used to decode a received point in a communications system.

24. (Original) The method of claim 23, wherein the communications system is an asymmetric digital subscriber line (ADSL) compliant system.

25 - 27. Cancelled.